



**KOCAELİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABORATUVARI-1  
PROJE -3  
AKILLI TELEFONLAR İLE NESNE TESPİTİ**

**ENGİN YENİCE  
190201133**

**KOCAELİ 2021**

# Akıllı Telefonlar ile Nesne Tespiti

## İçindekiler

React Native-APP.....	3
App.js.....	3
Home.js.....	3
ObjectDetection.js / Resim Seçme ve Resim Çekme.....	3
ObjectDetection.js / Resim Yükleme.....	4
Gallery.js / Firebase Veri Getirme.....	4
Gallery.js / Resim Görüntüle.....	4
PHP-API.....	6
Index.php.....	6
Application.php.....	6
Db.php.....	7
Storage.php.....	7

## React Native-APP

### App.js

1. Başla
2. SelectScreen adında bir useState oluştur.
3. SelectScreen == home ise
  1. Home.js çalıştır
4. SelectScreen == object ise
  1. ObjectDetection.js çalıştır
5. SelectScreen == gallery ise
  1. Gallery.js çalıştır
6. Bitir.

### Home.js

1. Başla
2. Ekranı Kocaeli Üniversitesi logosunu ve öğrenci bilgilerini yaz
3. Ekranı 2 adet buton ekle
4. Resim göster butonuna tıklanırsa
  1. SelectScreen = object yap
5. Tespit Edilen Resimler butonuna tıklanırsa
  1. SelectScreen = gallery yap
6. Bitir.

### ObjectDetection.js / Resim Seçme ve Resim Çekme

1. Başla
2. Ekranı 3 adet buton ekle (Anasayfa, Galeri, Kamera)
3. Galeri butonunu tıklanırsa
  1. Kullanıcının resim seçmesi için galeriyi aç.
  2. Resim seçilirse
    1. uploadImages metodunu çalıştır ve resmi API gönder
  3. Resim seçilmezse

1. İşlemi iptal et.
4. Kamera butonunu tıklanırsa
  1. Kullanıcının resim çekmesi için kamerayı aç.
  2. Resim onaylanırsa
    1. uploadImages metodunu çalıştır ve resmi API gönder
  3. Resim onaylanmazsa
    1. İşlemi iptal et.
5. Bitir.

## ObjectDetection.js / Resim Yükleme

1. Başla
2. Metoda parametre olarak gelen resim bilgilerini yeni oluşturulan FormData sınıfına ekle.
3. Global olarak belirtilen **FetchUrl** olarak belirtilen API adresine resim bilgilerini fetch kullanarak POST metodu ile gönder.
4. API tarafından geri döndürülen JSON verilerini gerekli değişkenlere aktar ve tespit edilen resmi ekranda göster
5. Bitir.

## Gallery.js / Firebase Veri Getirme

1. Başla
2. Global olarak belirtilen **FetchUrl** olarak isimlendirilen firebase veritabanına fetch sorgusu gönder. Gelen JSON datayı useState değişkenlerine aktar.
3. useState gelen resim linklerini for döngüsü ile dön ve ekrana küçük resim olarak ekle.
4. Bitir.

## Gallery.js / Resim Görüntüle

1. Başla
2. Galeri üzerinden herhangi resme tıklandığında

3. Modal verisini görünür yap ve resmi büyük boyutlarda göster
4. Bitir.

## PHP-API

### Index.php

1. Başla
2. \$\_FILES tipinde bir istek varsa
  1. Resmin ismini belirsiz bir isim ile değiştir.
  2. NormalImages klasörü içerisine resmi kayıt et.
  3. **application.php ObjectDetection** metodunu çağır. **ObjectDetection( Resim Global Yolu, Resim Dosya Yolu, Resim Tipi)**
  4. **db.php FirebaseAdd** metodunu çağır. **FirebaseAdd(ObjectDetection metodundan dönen JSON veri,Kayıt edilen resim)**
  5. FirebaseAdd metodundan dönen JSON datayı ekrana yaz.
3. \$\_FILES tipinde bir istek yoksa
  1. Ekrana JSON formatında sistem bilgilerini yaz.
4. Bitir.

### Application.php

1. Başla
2. VisionClient sınıfından bir nesne oluştur. Bu nesneyi oluştururken Auth işlemi yapmak için Google Cloud tarafından verilen json dosyasının yolunu belirt.
3. Metoda gönderilen resim yolunu kullanarak resmi aç.
4. Açılan resmi VisionClient sınıfının image metoduna gönder yanında da ek parametre olarak nesne tespiti için gerekli olan sabit değişkeni gönder. **OBJECT\_LOCALIZATION**
5. Dönen verileri düzenle
6. Gelen koordinat değerlerini x ve y olacak şekilde ayırmak için **AllCords** metoduna gönder.
7. Her bir nesnenin koordinatları arasından en büyük ve en küçük koordinatları almak için **MaxMinCords** metodunu çalıştır.
8. **ImageDraw** metodunu çalıştırarak resmin bir kopyasını oluştur ve koordinat bilgilerini üzerine çiz. Son olarak resmi detectedImages/ klasörü içerisine kayıt et.

9. **ImageTitlePush** metodunu çağırarak resim üzerinde tespit edilen nesneleri ve resmin renk kodlarının bulunduğu bir obje oluştur.
10. Oluşturulan obje ile resim url sinin bulunduğu objeleri birleştir ve **index.php**'ye geri döndür.
11. Bitir.

## Db.php

1. Başla
2. Parametre olarak gelen JSON data içerisinde bulunan resimlerinin adresini **storage.php**'ye gönder.
3. Parametre olarak gelen JSON data içerisinde gelen resim adreslerini **storage.php** tarafından gönderilen dosya adresleri ile değiştir.
4. Sunucu üzerinde bulunan resimleri sunucudan sil.
5. CURL işlemi ile verileri firebase realtime database'e kayıt et.
6. Bitir.

## Storage.php

1. Başla
2. Firebase auth işlemini gerçekleştirmek için gönderilen .json dosyasını Factory sınıfından türeyen nesneye ekle.
3. Sunucu üzerinde bulunan resmi Firebase Storage yükle.
4. Yüklenen resmin linkini geriye döndür.
5. Bitir.